



Docket No. 220480US2

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

IN RE APPLICATION OF: Hidehiko TANAKA, et al.

GAU: 2664

SERIAL NO: 10/092,989

EXAMINER: PHAM, BRENDA H

FILED: March 8, 2002

FOR: LOAD STORE QUEUE APPLIED TO PROCESSOR

SUBMISSION NOTICE REGARDING PRIORITY DOCUMENT(S)

COMMISSIONER FOR PATENTS
ALEXANDRIA, VIRGINIA 22313

SIR:

Certified copies of the Convention Application(s) corresponding to the above-captioned matter:

☒ are submitted herewith

☐ were filed in prior application filed

☐ were submitted to the International Bureau in PCT Application Number _____
Receipt of the certified copies by the International Bureau in a timely manner under PCT Rule 17.1(a) has been acknowledged as evidenced by the attached PCT/IB/304.

Respectfully Submitted,

OBLON, SPIVAK, McCLELLAND,
MAIER & NEUSTADT, P.C.


Marvin J. Spivak

Registration No. 24,913
Joseph Scafetta, Jr.
Registration No. 26,803

Customer Number

22850

Tel. (703) 413-3000
Fax. (703) 413-2220
(OSMMN 11/04)

日 本 国 特 許 庁
JAPAN PATENT OFFICE

別紙添付の書類に記載されている事項は下記の出願書類に記載されている事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed with this Office

出 願 年 月 日

Date of Application:

2001年 3月28日

出 願 番 号

Application Number:

特願2001-093994

ST.10/C]:

[JP2001-093994]

出 願 人
Applicant(s):

株式会社半導体理工学研究センター

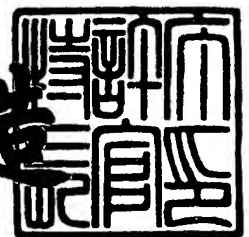
BEST AVAILABLE COPY

CERTIFIED COPY OF
PRIORITY DOCUMENT

2002年 3月29日

特許庁長官
Commissioner,
Japan Patent Office

及川耕造



出証番号 出証特2002-3022669

BEST AVAILABLE COPY

【書類名】 特許願

【整理番号】 A000007469

【提出日】 平成13年 3月28日

【あて先】 特許庁長官 殿

【国際特許分類】 G06F 12/08

【発明の名称】 ロード・ストアキュー

【請求項の数】 4

【発明者】

 【住所又は居所】 神奈川県横浜市港北区太尾町 1 4 2 3 - 4 1 9

 【氏名】 田中 英彦

【発明者】

 【住所又は居所】 茨城県筑波郡谷和原村絹の台 6 - 1 3 - 2 3

 【氏名】 坂井 修一

【発明者】

 【住所又は居所】 東京都荒川区西日暮里 1 - 2 - 7

 【氏名】 辻 秀典

【発明者】

 【住所又は居所】 千葉県船橋市前原東 6 - 2 2 - 4

 【氏名】 安島 雄一郎

【特許出願人】

 【識別番号】 396023993

 【氏名又は名称】 株式会社 半導体理工学研究センター

【代理人】

 【識別番号】 100058479

 【弁理士】

 【氏名又は名称】 鈴江 武彦

 【電話番号】 03-3502-3181

【選任した代理人】

 【識別番号】 100084618

【弁理士】

【氏名又は名称】 村松 貞男

【選任した代理人】

【識別番号】 100068814

【弁理士】

【氏名又は名称】 坪井 淳

【選任した代理人】

【識別番号】 100092196

【弁理士】

【氏名又は名称】 橋本 良郎

【選任した代理人】

【識別番号】 100091351

【弁理士】

【氏名又は名称】 河野 哲

【選任した代理人】

【識別番号】 100088683

【弁理士】

【氏名又は名称】 中村 誠

【手数料の表示】

【予納台帳番号】 011567

【納付金額】 21,000円

【提出物件の目録】

【物件名】 明細書 1

【物件名】 図面 1

【物件名】 要約書 1

【包括委任状番号】 9810816

【プルーフの要否】 要

【書類名】 明細書

【発明の名称】 ロード・ストアキュー

【特許請求の範囲】

【請求項 1】 ストア命令のイン・オーダー情報を格納する第 1 のテーブルと、

ストア命令のストア・アドレスとストア値を保持し、アドレスをインデックスとして検索することが可能な第 2 のテーブルと、

ロード命令のタグ情報に応じて、前記第 1 のテーブル内に格納されたストア命令から、前記ロード命令に先行するストア命令を検索する第 1 の検索手段と、

前記ロード命令のアドレスに応じて前記第 2 のテーブルから読み出されたストア命令と前記第 1 の検索手段により検索されたストア命令が一致した場合、一致信号を出力する第 2 の検索手段と

を具備することを特徴とするロード・ストアキュー。

【請求項 2】 前記第 1 のテーブル内のキューの先頭を示す第 1 のポインタと、前記第 1 のテーブル内のキューの末尾を示す第 2 のポインタをさらに具備することを特徴とする請求項 1 記載のロード・ストアキュー。

【請求項 3】 前記第 2 のテーブルは、前記ロード命令のアドレスに応じてストア値を出力することを特徴とする請求項 1 記載のロード・ストアキュー。

【請求項 4】 リタイアするストア命令のタグ情報に応じて、前記第 1 のテーブルのエントリを削除するとともに、前記リタイアするストア命令のアドレスに応じて前記第 2 のテーブルからストア・アドレスとストア値を読み出すリタイア処理手段をさらに具備することを特徴とする請求項 1 記載のロード・ストアキュー。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】

本発明は、例えばプロセッサに適用されるロード・ストアキューに関する。

【0002】

【従来の技術】

プロセッサにおける命令の実行手順の1つとして、アウト・オブ・オーダー (out-of-order) 実行がある。このアウト・オブ・オーダー実行は先行命令と依存関係にない後続命令を順不同に実行する方式である。

【0003】

ロード・ストアキューは、アウト・オブ・オーダー実行を行うプロセッサにおいて、アウト・オブ・オーダーに処理されるロード・ストア命令に対して、メモリを介したデータ依存関係を維持するために利用される。

【0004】

具体的には、ロード・ストアキューは、メモリアクセス命令の順番を把握し、アウト・オブ・オーダーに発行されるメモリアクセス命令の順序を管理する。したがって、アウト・オブ・オーダー実行によるメモリアクセスの追い越しを回避できる。

【0005】

図2は、ロード・ストアキューとプロセッサ及びデータ・キャッシュとの関係を示している。

【0006】

図2において、ロード・ストアキュー11は、プロセッサ（プロセッサの実行機構）12と、データ・キャッシュ13の間に配置されている。このロード・ストアキュー11は、プロセッサからアウト・オブ・オーダーに発行されるストア要求とロード要求を全て受取り、イン・オーダー (in-order) を保証したストア命令をデータ・キャッシュ13に書き込み、イン・オーダーを保証したロード値をマルチプレクサ (MUX) 14を介してプロセッサ（プロセッサの実行ユニット）11に返す。ロード・ストアキュー11は、アウト・オブ・オーダーに発行されるストア命令を保持する図示せぬテーブルと、アウト・オブ・オーダーに発行されるロード命令の値を読み出し、選択する機構により構成されている。

【0007】

次に、図2に示すロード・ストアキュー、プロセッサ及びデータ・キャッシュの動作について概略的に説明する。ロード・ストアキューの動作は次の4つに分けられる。

【 0 0 0 8 】

1. (イン・オーダー情報の取得処理)

プロセッサ 1 2 により、ストア命令がデコードされた時点（アウト・オブ・オーダーに処理される前）において、ロード・ストアキュー 1 1 はプロセッサ 1 2 からイン・オーダーにストア命令のデコード情報（2 1）を受け取る。これにより、ストア命令のイン・オーダー情報を確保する。

【 0 0 0 9 】

2. (ストア要求の処理)

プロセッサ 1 2 内で、ストア命令のアドレスと値が確定した時点で、ロード・ストアキュー 1 1 はプロセッサ 1 2 からストア要求（2 2）を受け取り、ロード・ストアキュー 1 1 の中にストア命令のアドレスと値を保持する。この値は、ストア命令がアウト・オブ・オーダーの状態の間、つまり、ストア命令がリタイアしてイン・オーダーが確定するまで保持される。

【 0 0 1 0 】

3. (ロード要求の処理)

ロード・ストアキュー 1 1 は、プロセッサ 1 2 からのロード要求（2 5）に応じて、ロード命令のアドレスを受け取り、ロード命令を処理する。仮に、このロード命令がロード・ストアキュー 1 1 内にあるストア値（イン・オーダーが確定してない当該ロード命令に先行するストア値）に対するものであった場合、ロード・ストアキュー 1 1 の当該ロード命令の値を持つストア命令から値を取り出す。該当するストア命令がなければデータ・キャッシュ 1 3 よりロード値（2 6）を読み込む。

【 0 0 1 1 】

実際の動作において、プロセッサ 1 2 は、該当するロード要求（2 5）をデータ・キャッシュ 1 3 とロード・ストアキュー 1 1 に対して発行する。ロード・ストアキュー 1 1 にこのロード命令に先行するストア値がある場合、ロード・ストアキュー 1 1 からデータが存在する旨を示すヒット信号（2 8）がマルチプレクサ 1 4 に供給される。マルチプレクサ 1 4 はヒット信号（2 8）が供給された場合、ロード・ストアキュー 1 1 からのロード値（2 7）を選択する。また、先行

するストア値がなくヒット信号（28）が供給されない場合、マルチプレクサ14は、データ・キャッシュ13からのロード値（26）を選択する。このようにしてマルチプレクサ14により選択されたロード値（29）はプロセッサ12に供給される。

【0012】

4.（ストア命令のリタイア処理）

ストア命令が確定し、イン・オーダー順が確定した場合、プロセッサ（プロセッサの実行機構）12からリタイア情報（23）が出力される。ロード・ストアキュー11はリタイア情報（23）を受け取り、ロード・ストアキュー11内にあるストア命令をデータ・キャッシュ13に書き戻し（24）、ロード・ストアキュー11内のエントリを削除する。

【0013】

図3は、従来のロード・ストアキューの構成を示している。従来のロード・ストアキューは、タグ情報及びアドレスにより連想検索可能なキューとして実装されている。

【0014】

すなわち、図3に示すロード・ストアキュー11は、例えば連想検索が可能なテーブル30と、キューの先頭を表すポインタ31と末尾を表すポインタ32、複数のセクタ33、34、35、36、比較器37、38、39、40及びマルチプレクサ41により構成されている。

【0015】

上記構成において、前述した4つの動作について説明する。

【0016】

1.（イン・オーダー情報の取得処理）

この処理は、デコード時のイン・オーダー情報として、ストア命令のタグ情報（イン・オーダーに与えられるプロセッサ内で一意な番号）を受け取り、キュー内にイン・オーダーにエントリを確保する。

【0017】

すなわち、先ず、デコード時のイン・オーダー情報としてストア命令のタグ情

報（ストアタグ（51））がプロセッサから出力される。このストアタグ（51）は、セクタ34に供給される。

【0018】

ポインタ32はキューの末尾に対応するテーブル30のエントリを指定する（52）。セクタ34はポインタ32により指定されたエントリにタグ情報を供給する。このタグ情報は指定されたエントリに書き込まれ、これとともに、バリッドが有効とされる（53）。

【0019】

このとき、ロード・ストアキュー11のストア命令のアドレス、ストア値のエントリは空のままである。また、ポインタ32が指し示すエントリは、ストアタグ（51）の入力タイミングに応じて例えば“+1”されて変更される。

【0020】

2. （ストア要求の処理）

この処理は、ストア命令のタグ情報、ストア命令のアドレス、ストア値を受け取り、すでに確保されている（タグ情報が一致する）エントリに書き込む。

【0021】

すなわち、ストア要求として、ストア命令のタグ情報（ストアタグ（54））、ストア命令のアドレス（ストア・アドレス（55））、ストア値（56）がプロセッサから出力される。前記ストアタグ（54）は比較器37に供給され、ストア・アドレス（55）、ストア値（56）はセクタ35、36にそれぞれ供給される。

【0022】

比較器37はテーブル30のバリッドエントリを検索して有効なタグを取り出す（57）。これとともに、テーブル30のタグエントリからストアタグ（54）と一致するエントリを検索する（58）。このようにして、ストア・アドレス（55）、ストア値（56）の書き込み先エントリを検索する。

【0023】

セクタ35、36は、検索されたエントリ（59）にストア・アドレス（55）及びストア値（56）を供給する。このようにして、テーブル30に確保さ

れているエントリにストア・アドレス（55）、ストア値（56）が書き込まれる。

【0024】

3.（ロード要求の処理）

この処理は、ロード命令のタグ情報と、ロード命令のアドレスを受け取り、先行するストア命令（タグ情報により検索）のうちアドレスが一致する（アドレスにより検索）ストア命令を検索する。該当するストア命令が存在する場合、その旨を知らせるヒット信号を出力するとともに、ロード値としてアドレスが一致するストア命令の値を出力する（ストアからロードへの値のフォワーディング）。

【0025】

すなわち、ロード要求として、ロード命令のタグ情報（ロードタグ（62））及びロード命令のアドレス（ロード・アドレス（63））がプロセッサより出力される。比較器38、39はロードタグ（62）及びロード・アドレス（63）を受け取る。

【0026】

比較器38は、テーブル30のバリッドエントリを検索して有効なタグを取り出す（64）。これと同時に、テーブル30のタグエントリを検索することにより、ロードタグ（62）に先行した位置にあるストア命令（65）を検索する。

【0027】

比較器39は、テーブル30のアドレス・エントリを検索してロード・アドレス（63）と一致するストア命令のエントリを検索する（66）。

【0028】

比較器40は、比較器38、39の出力信号に基づき、ロード命令に先行し、かつアドレスが一致するストア命令のエントリが存在するかどうか調べる。この結果、このようなストア命令が存在する場合、ヒット信号（67）を出力する。さらに、マルチプレクサ41は、比較器40の出力信号に応じて該当するエントリを選択し、そのエントリの値をロード値（68）として出力する。

【0029】

4.（ストア命令のリタイア処理）

この処理は、リタイア情報として、リタイアするストア命令のタグ情報を受け取り、該当するストア命令のエントリをキューから削除する。

【0030】

すなわち、リタイア情報として、リタイアするストア命令のタグ情報（ストアタグ（69））がプロセッサから出力される。セクタ33はこのストアタグ（69）を受ける。

【0031】

ポインタ31はキューの先頭に対応する削除すべきテーブルのエントリを指定する（70）。前記セクタ33はポインタ31により指定されたエントリにストアタグ（69）を供給する。これに応じて、テーブル30の指定されたエントリのストア・アドレスと値はデータ・キャッシュに出力され（71）、そのエントリが削除される。前記ポインタ31が指し示すエントリは前記ストアタグ（69）の入力タイミングに応じて例えば“+1”され、変更される。

【0032】

【発明が解決しようとする課題】

ところで、上記従来のロード・ストアキューは、ロード・ストアキューに対するストア要求処理の際、イン・オーダーに確保されたテーブル30に対し、ストア命令のアドレスと値をテーブル30の該当エントリに書き込む必要がある。このため、ストア命令のアドレスと値を書き込むエントリを検索するために、テーブル30の全タグエントリがストアタグに一致しているかどうかを検索して、書き込み先を特定しなければならない。このため、検索に長時間を要し、高速処理が困難であった。

【0033】

また、ロード・ストアキューに対するロード要求処理の際、ロード命令に先行するストア命令を検索するとともに、ロード命令とアドレスが一致する命令を検索する必要がある。すなわち、従来は、先行するストア命令のエントリを読み出した後、その中からアドレスが一致するものを検索する必要がある。したがって、ロード要求処理においても、処理の高速化が困難であった。

【0034】

さらに、ストア要求処理とロード要求処理において、ストア・アドレス、ロード・アドレスにより、テーブル 30 の全エントリを検索する必要がある。このため、大規模なエントリを有するテーブルを構成しようとした場合、処理速度が低下するとともに、回路構成が複雑化し実現が困難であった。

【0035】

本発明は、上記課題を解決するためになされたものであり、その目的とするところは、高速処理が可能であるとともに、大規模なエントリを有するテーブルを小規模な回路により構成することが可能なロード・ストアキューを提供しようとするものである。

【0036】

【課題を解決するための手段】

本発明のロード・ストアキューは、上記課題を解決するため、ストア命令のイン・オーダー情報を格納する第 1 のテーブルと、ストア命令のストア・アドレスとストア値を保持し、アドレスをインデックスとして検索することが可能な第 2 のテーブルと、ロード命令のタグ情報に応じて、前記第 1 のテーブル内に格納されたストア命令から、前記ロード命令に先行するストア命令を検索する第 1 の検索手段と、前記ロード命令のアドレスに応じて前記第 2 のテーブルから読み出されたストア命令と前記第 1 の検索手段により検索されたストア命令が一致した場合、一致信号を出力する第 2 の検索手段とを具備している。

【0037】

【発明の実施の形態】

以下、本発明の実施の形態について図面を参照して説明する。

【0038】

図 1 は、本発明に係るロード・ストアキューの一実施形態を示している。

【0039】

図 3 に示す、従来のロード・ストアキュー 11 は、ストア命令のイン・オーダー情報とストア命令のアドレスと値を同じテーブル 30 により管理していた。これに対して、本発明のロード・ストアキューは、ストア命令のイン・オーダー情報を格納するテーブルと、ストア命令のアドレスと値を保持するテーブルを分離

している。

【0040】

すなわち、図1に示す本発明のロード・ストアキュー100は、イン・オーダー・ステイトキュー101、テンポラル・ストアキャッシュ102、ポインタ103、ポインタ104、セクタ105、106、及び比較器107、108を有している。

【0041】

前記イン・オーダー・ステイトキュー101は、タグにより連想検索が可能なテーブルである。イン・オーダー・ステイトキュー101は、複数のタグエントリと、これらタグエントリに対応されたバリッドエントリを有している。但し、イン・オーダー・ステイトキュー101の構成はこれに限定されるものではない。このイン・オーダー・ステイトキュー101は、プロセッサにより、例えばロード命令、或いはストア命令がデコードされた時点でエントリが確保される。

【0042】

一方、前記テンポラル・ストアキャッシュ102はアドレスをインデックスとして検索することが可能なテーブルである。このこのテンポラル・ストアキャッシュ102は、複数のアドレス・エントリと、これらアドレス・エントリに対応されたバリッドエントリ、タグエントリ、バリューエントリを有している。ストア命令は、例えばアドレスが計算された時点で、ストア・アドレスを用いてテンポラル・ストアキャッシュ102にストア値が書き込まれる。

【0043】

前記ポインタ103は、イン・オーダー・ステイトキュー101内の先頭、すなわち、実行中のロード・ストア命令の中で最も古いものを指し示している。ポインタ104は、イン・オーダー・ステイトキュー101内の末尾、すなわち、実行中のロード・ストア命令の中で最も新しいものを指し示している。前記セクタ105は、例えばリタイア情報を受け、セクタ106は、例えばデコード情報を受ける。

【0044】

比較器107は、イン・オーダー・ステイトキュー101に対応して配置され

ている。この比較器 1 0 7 には、ロード要求処理時に、ロード命令のタグ情報（ロードタグ）が供給される。この比較器 1 0 7 は、供給されたロードタグに応じて、このロード命令に先行するストア命令がイン・オーダー・ステイトキュー 1 0 1 内にあるかどうかを検索する。この結果、先行するストア命令がある場合、そのストア命令のタグ情報をターゲットタグとして出力する。

【 0 0 4 5 】

比較器 1 0 8 は、テンポラル・ストアキャッシュ 1 0 2 に対応して配置されるともに、前記比較器 1 0 7 に接続されている。この比較器 1 0 8 は、テンポラル・ストアキャッシュ 1 0 2 内のタグを検索するとともに、検出したタグが比較器 1 0 7 から供給されるターゲットタグと一致するかどうかを比較する機能を有している。

【 0 0 4 6 】

次に、上記構成のロード・ストアキューの動作について説明する。

【 0 0 4 7 】

1. (イン・オーダー情報の取得)

デコード時のイン・オーダー情報としてストア命令のタグ情報（ストアタグ（1 2 1））がプロセッサから出力される。このストアタグ（1 2 1）は、セレクタ 1 0 6 に供給される。

【 0 0 4 8 】

ポインタ 1 0 4 は、イン・オーダー・ステイトキュー 1 0 1 の末尾のエントリを示しており、このポインタ 1 0 4 により、イン・オーダー・ステイトキュー 1 0 1 を書き込むエントリが指定される（1 2 2）。

【 0 0 4 9 】

セレクタ 1 0 6 は、ポインタ 1 0 4 により指定されたイン・オーダー・ステイトキュー 1 0 1 の末尾のエントリに、前記入力されたストアタグ（1 2 1）を供給する。このストアタグ（1 2 1）は指定されたエントリに書き込まれ、このストアタグに対応するバリッドが有効とされる（1 2 3）。

【 0 0 5 0 】

前記ポインタ 1 0 4 の内容は、前記ストアタグ（1 2 1）の入力タイミングに

応じて例えば“+1”され、変更される。

【0051】

2. (ストア要求の処理)

次に、ストア要求として、ストア命令のタグ情報(ストアタグ(125))、ストア命令のアドレス(ストア・アドレス(124))、ストア値(126)がプロセッサから出力される。

【0052】

前述したように、テンポラル・ストアキャッシュ102は、アドレスをインデックスとしてエントリを指定することが可能なテーブルである。すなわち、テンポラル・ストアキャッシュ102のエントリは、ストア・アドレス(124)をインデックスとして指定され、この指定されたエントリにストアタグ(125)、ストア値(126)が書き込まれる。

【0053】

3. (ロード要求の処理)

次に、ロード要求として、ロード命令のタグ情報(ロードタグ(127))及びロード命令のアドレス(ロード・アドレス(128))がプロセッサより出力される。前記ロードタグ(127)は比較器107に供給される。この比較器107は、このロードタグ(127)に応じて、先行するストア命令を検索する。すなわち、比較器107は、イン・オーダー・ステイトキュー101のバリッドエントリを検索して有効なタグ情報を取り出す(129)。これとともに、このタグ情報に従ってタグエントリを検索することにより、入力されたロードタグ(127)に先行する位置にあるストア命令を検索(130)する。この検索されたストア命令はターゲットタグ(131)として比較器108に供給される。

【0054】

一方、前記ロード・アドレス(128)はテンポラル・ストアキャッシュ102にインデックスとして供給される。テンポラル・ストアキャッシュ102は、インデックスとしてのロード・アドレス(128)と一致する有効なストアタグ(132)と、これに対応する値を読み出す。この読み出されたストアタグ(132)とバリッドデータ(133)は、比較器108に供給される。

【 0 0 5 5 】

比較器 1 0 8 は、テンポラル・ストアキャッシュ 1 0 2 から供給されたストアタグ (1 3 2) が、前記比較器 1 0 7 から供給されるロード命令に先行するストア命令のタグと一致するかどうかを比較する。この結果、これらが一致する場合、比較器 1 0 8 はヒット信号を出力 (1 3 4) を出力する。これとともに、前記テンポラル・ストアキャッシュ 1 0 2 から読み出されたストア値は、ロード値 (1 3 5) として出力される。

【 0 0 5 6 】

4. (ストア命令のリタイア処理)

ストア命令のリタイア処理において、先ず、リタイアするストア命令のタグ情報 (ストアタグ) 及びストア・アドレス (1 3 6) がプロセッサから出力される。セレクタ 1 0 5 はこのストアタグ及びストア・アドレス (1 3 6) を受ける。

【 0 0 5 7 】

ポインタ 1 0 3 はキューの先頭に対応するイン・オーダー・ステイトキュー 1 0 1 のエントリを指定する。前記セレクタ 1 0 5 はポインタ 1 0 3 により指定されたエントリに前記ストアタグ (1 3 6) を供給する。これに応じて、指定されたバリッドエントリ及びタグエントリを削除する (1 3 7) 。

【 0 0 5 8 】

さらに、セレクタ 1 0 5 は入力された前記ストア・アドレス (1 3 6) をインデックスとして、テンポラル・ストアキャッシュ 1 0 2 のエントリを指定し (1 3 8) 、この指定したエントリのストア・アドレスとストア値を図示せぬデータ・キャッシュに出力する (1 3 9) 。それとともに、そのエントリの内容を削除する。

【 0 0 5 9 】

また、前記ポインタ 1 0 3 の内容は、前記ストアタグ (1 3 6) の入力タイミングに応じて例えば “ + 1 ” され、変更される。

【 0 0 6 0 】

上記実施形態によれば、ストア命令のイン・オーダー・情報をイン・オーダー・ステイトキュー 1 0 1 に格納し、ストア命令のアドレス及びストア値は、アド

レスをインデックスとするテンポラル・ストアキャッシュ 1 0 2 に格納している。このため、ストア処理要求を高速に処理できる。すなわち、従来の方式では、書き込むエントリを検索するために、テーブルの全タグエントリがストアタグに一致しているかどうかを検索し、書き込み先を特定する必要を有していた。これに対して、本実施形態の場合、ストア・アドレスをインデックスとしてテンポラル・ストアキャッシュ 1 0 2 を検索でき、ストアタグの検索なしにエントリを特定できる。したがって、高速にストア処理を実行できる。

【 0 0 6 1 】

また、本実施形態によれば、ロード要求処理において、ロード命令とアドレスが一致するストア命令の検索を従来に比べて高速化できる。すなわち、従来の方式では、先行するストア命令のエントリを読み出し、この読み出したエントリの中からアドレスが一致するものを検索していたため処理に長時間を要していた。これに対して、本実施形態の場合、ロード要求処理において、先行するストア命令の検索と、アドレスが一致するストア命令の読み出しを並行して実行でき、両者が一致するかどうかを比較するだけでロード命令とアドレスが一致するストア命令を検索することができる。したがって、高速な処理が可能である。

【 0 0 6 2 】

しかも、テンポラル・ストアキャッシュ 1 0 2 は、アドレスをインデックスとしてアクセスできるため、ストア・アドレスによりストアタグを検索することなくエントリを特定できる。したがって、高速な処理が可能である。

【 0 0 6 3 】

また、従来の構成の場合、ストア要求とロード要求の際にアドレスによるエントリの全検索が必要である。このため、ロード・ストアキューのエントリを増やした場合処理速度が低下するため、ロード・ストアキューのエントリを増化することが困難である。これに対して、本実施形態の場合、テンポラル・ストアキャッシュ 1 0 2 はアドレスをインデックスとしたテーブルにより構成されている。このため、従来のように、アドレスによるエントリの全検索が不要であるため、テンポラル・ストアキャッシュ 1 0 2 の大規模化が可能であり、規模が従来と同様であれば、従来の方式より高速化が可能である。

【 0 0 6 4 】

さらに、従来の方式の場合、先行するストア命令のエントリを読み出し、この読み出したエントリの中からアドレスが一致するものを検索していた。このため多くの比較器を必要としていた。これに対して、本実施形態の場合、ロード要求処理において、先行するストア命令をイン・オーダー・ステイトキュー 1 0 1 から検索するための比較器 1 0 7、及びこの比較器 1 0 7 からの出力と、テンポラル・ストアキャッシュ 1 0 2 から読み出したストア命令とを比較する比較器 1 0 8 だけでよい。このため、比較器の数を削減することができ回路構成を簡単化できる利点を有している。

【 0 0 6 5 】

尚、ストア命令のイン・オーダー情報を格納するイン・オーダー・ステイトキュー 1 0 1 の構成は、イン・オーダー情報が検索できるテーブルであれば、どのような形でもよい。

【 0 0 6 6 】

また、ストア値を保持するテンポラル・ストアキャッシュ 1 0 2 は、アドレスをインデックスとしているが、アドレス空間の広さと用意するエントリによって、アドレス・エントリのマッピング方法は任意に変えてもよい。このため、テンポラル・ストアキャッシュ 1 0 2 は、アドレスをインデックスとした従来の殆どのキャッシュの構成を適用できる。

【 0 0 6 7 】

その他、本発明の要旨を変えない範囲において種々変形実施可能なことは勿論である。

【 0 0 6 8 】

【発明の効果】

以上、詳述したように本発明によれば、高速処理が可能であるとともに、大規模なエントリを有するテーブルを小規模な回路により構成することが可能なロード・ストアキューを提供できる。

【図面の簡単な説明】

【図 1】

本発明の実施形態を示す構成図。

【図 2】

ロード・ストアキューとプロセッサ及びデータ・キャッシュとの関係を示す構成図。

【図 3】

従来のロード・ストアキューを示す構成図。

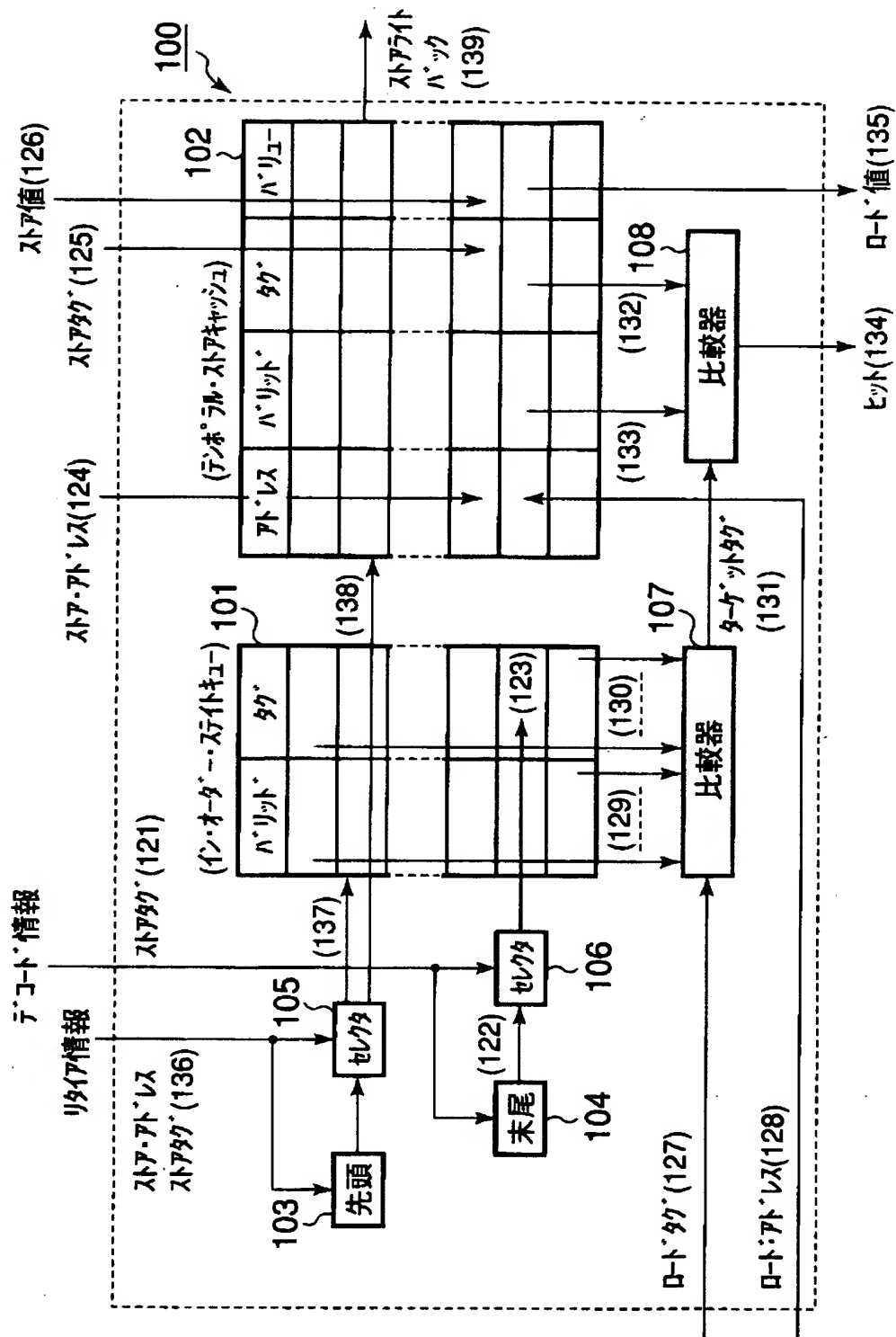
【符号の説明】

- 1 0 0 …ロード・ストアキュー、
- 1 0 1 …イン・オーダー・ステイトキュー、
- 1 0 2 …テンポラル・ストアキャッシュ、
- 1 0 3、1 0 4 …ポインタ、
- 1 0 5、1 0 6 …セレクタ、
- 1 0 7、1 0 8 …比較器。

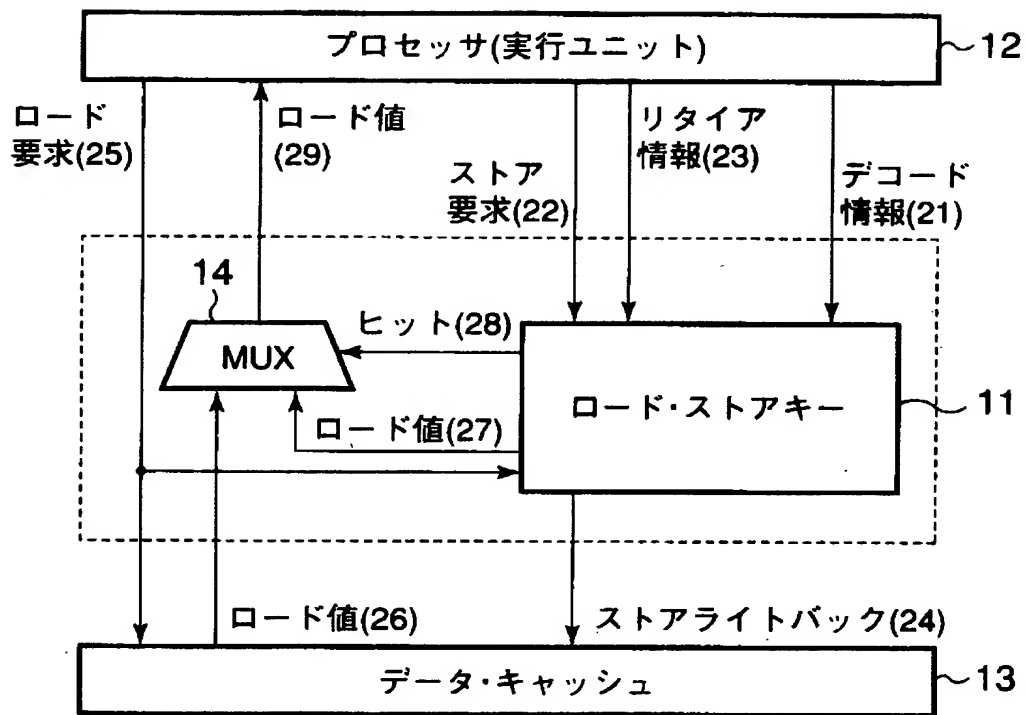
【書類名】

図面

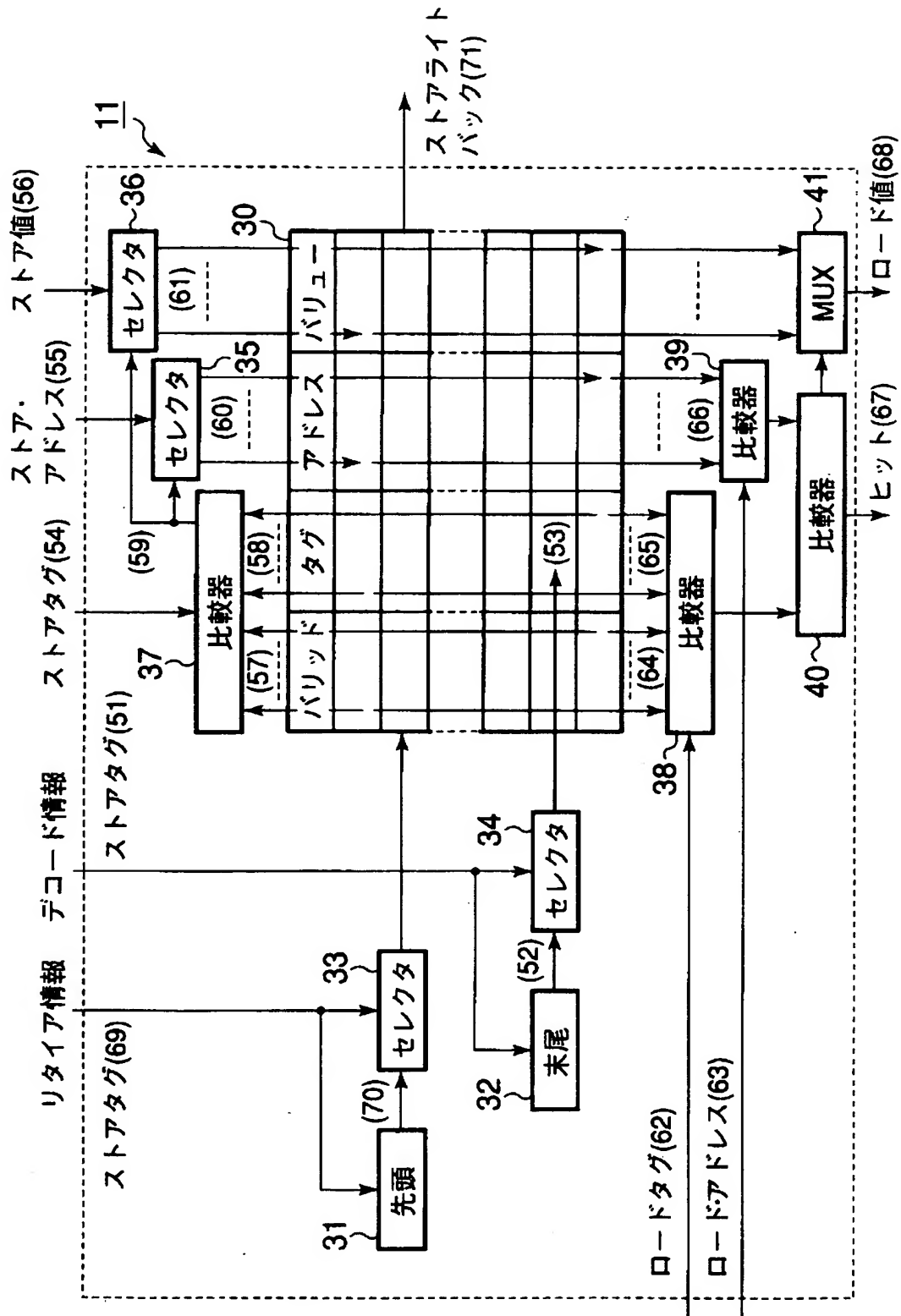
【図 1】



【図 2】



【図 3】



【書類名】 要約書

【要約】

【課題】 高速処理が可能であり、大規模なエントリを有するテーブルを小規模な回路により構成することが困難であった。

【解決手段】 ストア命令のイン・オーダー・情報をイン・オーダー・ステイトキュー 1 0 1 に格納し、ストア・アドレス及びストア値はアドレスをインデックスとするテンポラル・ストアキャッシュ 1 0 2 に格納している。このため、ロード命令とアドレスが一致するストア命令の検索を従来に比べて高速化できる。

【選択図】 図 1

出 願 人 履 歴 情 報

識別番号 [396023993]

1. 変更年月日 2001年 3月23日

[変更理由] 住所変更

住 所 神奈川県横浜市港北区新横浜3丁目17番地2 友泉新横浜ビル6階

氏 名 株式会社半導体理工学研究センター